

2. Crie uma classe abstrata de nome círculo. Nesta classe:

- Crie um atributo privado raio e respetiva propriedade;
- Crie o método abstrato area;
- Implemente o método perimetro que permite calcular o perímetro do círculo.

3. Crie uma classe de nome cilindro, derivada da classe abstrata círculo e, nessa classe:

- Crie o atributo privado altura e respetiva propriedade;
- Implemente o método area que permite calcular a área;
- Implemente o método volume que permite calcular o volume do cilindro;
- Implemente o método area_lateral que permite calcular a área lateral do cilindro;
- Implemente o método areaTotal que permite calcular a área total do cilindro.

4. Crie uma classe de nome esfera, derivada da classe abstrata círculo e, nessa classe implemente o método area. Este método permite calcular a área da esfera.

```
from abc import ABC, abstractmethod
from math import pi
```

```
# Classe abstrata Círculo
```

```
class Circulo(ABC):
```

```
    def __init__(self, raio):
        self.__raio = raio
```

```
    @property
```

```
    def raio(self):
        return self.__raio
```

```
    @raio.setter
```

```
    def raio(self, valor):
        if valor > 0:           #Extra
            self.__raio = valor
```

```
    else:
```

```
        raise ValueError("O raio deve ser maior que zero.")
```

```
    @abstractmethod #Ao criar um método abstracto ele não contém nada
```

`@abstractmethod` não cria um método abstrato, ele nos permite criar

```
def area(self):  
    pass  
  
def perimetro(self):  
    return 2 * pi * self.raio
```

Classe derivada Cilindro

```
class Cilindro(Circulo):  
    def __init__(self, raio, altura):  
        super().__init__(raio)  
        self.__altura = altura  
  
    @property  
    def altura(self):  
        return self.__altura  
  
    @altura.setter  
    def altura(self, valoraltura):  
        if valoraltura > 0:          #Extra  
            self.__raio = valoraltura  
        else:  
            raise ValueError("A altura deve ser maior que zero.")  
  
    def area(self):  
        # Área da base:  $\pi r^2$   
        return pi * (self.raio **2 )  
  
    def volume(self):  
        # Volume do cilindro:  $\pi r^2 h$   
        return pi * (self.raio **2 ) * self.altura  
        #return self.area()*self.altura  
  
    def area_lateral(self):  
        # Área lateral:  $2\pi rh$   
        return 2 * pi * self.raio * self.altura  
  
    def area_total(self):  
        # Área total:  $2\pi r^2 + 2\pi rh$   
        return self.area()
```

```

# Classe derivada Esfera (de Círculo)
class Esfera(Circulo):
    def area(self):
        # Área da esfera:  $4\pi r^2$ 
        return 4 * pi * self.raio**2

    def volume(self):
        # Volume da esfera:  $(4/3)\pi r^3$ 
        return (4 / 3) * pi * self.raio**3

    def diametro(self):
        # Diâmetro da esfera:  $2r$ 
        return 2 * self.raio

# Inserir valores
raio = float(input("Digite o valor do raio: "))
altura = float(input("Digite o valor da altura (para o cilindro): "))

# Criar objetos
#circulo = Cilindro(raio, altura)
esfera = Esfera(raio)
cilindro = Cilindro(raio, altura)

# Apresentar resultados
print("\n--- Resultados ---")
#print(f"Área do círculo: {circulo.area():.2f}")
#print(f"Perímetro do círculo: {circulo.perimetro():.2f}")
print(f"Área do círculo: {cilindro.area():.2f}")
print(f"Perímetro do círculo: {cilindro.perimetro():.2f}")
print(f"Área da esfera: {esfera.area():.2f}")
print(f"Perímetro da esfera: {esfera.perimetro():.2f}")
print(f"Volume da esfera: {esfera.volume():.2f}")
print(f"Diâmetro da esfera: {esfera.diametro():.2f}")
print(f"Área lateral do cilindro: {cilindro.area_lateral():.2f}")
print(f"Área total do cilindro: {cilindro.area_total():.2f}")
print(f"Volume do cilindro: {cilindro.volume():.2f}")

```



Digite o valor do raio: 4
Digite o valor da altura (para o cilindro): 8

--- Resultados ---

Área do círculo: 50.27
Perímetro do círculo: 25.13
Área da esfera: 201.06
Perímetro da esfera: 25.13
Volume da esfera: 268.08
Diâmetro da esfera: 8.00
Área lateral do cilindro: 201.06
Área total do cilindro: 50.27
Volume do cilindro: 402.12